GO CORPORATION
950 TOWER LANE
14TH FLOOR
FOSTER CITY, CA
94404

Background Information
GO Corp.'s PenPoint™ Operating System
For Mobile, Pen-Based Computers

Contacts:

Carol Broadbent (415) 345-7400
GO Corp.
950 Tower Lane, Suite 1400
Foster City, Calif. 94404

Vicky Hastings (503) 221-1063
Hastings, Humble, Giardini & Freeman Inc.
1200 N.W. Front Avenue, Suite 310
Portland, Ore. 97209

Julie McHenry (415) 592-7600
Wilson McHenry Company
255 Shoreline Drive, Suite 520
Redwood City, Calif. 94065

Contents

A New Market With Unique Requirements 1
Why a New Operating System? 1
PenPoint: Designed for the Pen and Mobility 2
The NUI Delivers a New Level of Ease of Use 3
Simple Navigation and Intuitive Organization 3
A Table of Contents Instead of File Directories and Windows 4
Gestures 4
The Directness of the Pen 5
Handwriting Translation 6
Embedded Document Architecture 8
Live Document Embedding 9
Hyperlinks 9
PenPoint Delivers Mobility 9
Instant On 9
Deferred Input and Output 10
Detachable Connectivity 11
Connecting PenPoint to Other Systems 11
PenPoint is Compact and Scalable 12
Device Independence 12
Scalable User Interface 12
No Keyboard Required 13
Memory Conservation 13
Different Memory Configurations 13
An Operating System for the 1990s 14
Summary 14
Glossary 16

The PenPoint™ operating system, from GO Corporation, is the only general purpose operating system designed for mobile, pen-based computers. PenPoint delivers major new benefits to users and developers of pen-based applications. These innovations include:

- the Notebook User Interface, or NUI
- the Embedded Document Architecture (EDA™)
- features to support mobile connectivity
- a compact and scalable implementation
- a modern, object-oriented design for the 1990s

A New Market With Unique Requirements

When GO Corp. founders established the company in 1987, they had one goal: to create a market for mobile, pen-based computers. In exploring the needs of this market, GO quickly discovered that mobile, pen-based computing represented a new market, distinct from the existing desktop computer market. Many of the users and applications would be different from the desktop environment.

By virtue of their form factor and lack of a keyboard, mobile, pen-based computers can be used in new settings where desktop or portable computers have in the past been inappropriate or impractical. They are particularly suitable for face-to-face activities, such as meetings and customer visits, where keyboards can be disruptive or socially unacceptable. Since pen-based computers do not require typing, they can also be used while walking, standing or moving about, allowing a machine to be held in one hand and operated with a pen held in the other.

Pen-based computers should fade into the background and be as unobtrusive as possible, not just to the user, but also to those with whom the user is working. Many mobile, pen-based users will use their computers to support other tasks that demand their concentration. For example, an insurance salesperson needs to concentrate on a client, not on a computer.

Because of the way in which mobile, pen-based computers will be used, a general purpose operating system for pen-based computing must meet two key requirements: it must provide unparalleled ease of use, and it must provide support for mobile connectivity so users can move from one work site to another without restrictions. In addition, it must provide a rich development environment suitable for developing a wide range of vertical and horizontal applications.

Why a New Operating System?

In investigating operating systems for mobile, pen-based computers, GO's founders quickly realized that existing systems were not suitable for the kind of mobile, easy-to-use, compact computing systems its founders envisioned. GO faced three choices:

- Build a "pen compatibility layer" onto an existing operating system.
- Merge selected components of several existing operating systems to create a new mobile, pen-based system.

- Create a new operating system from the ground up.

The first approach, layering a pen interface on top of an existing system, was rejected because it was likely to significantly compromise the ease of use promised by the pen. Layering a pen on top of an existing system doesn't, by itself, make a computer easy to use, just as a mouse doesn't guarantee ease of use. In fact, the layered approach could well increase the complexity of the user interface.

With a layered approach, the pen becomes an alternate form of keyboard and mouse. Because pen events are mapped onto a keyboard and mouse, users must select a mode by pressing a button on the pen or screen. To emulate the special control characters used by desktop systems, users must have a way of entering them by tapping with the pen on a picture of a keyboard. Furthermore, a pen interface, by itself, does nothing to simplify the complexities associated with desktop operating systems such as starting applications, opening and saving files, working with directories, and so on.

Because many mobile, pen-based applications will be fundamentally different from desktop applications — for example, forms completion, electronic calendars, freehand sketching and notetaking — the benefit of using an existing operating system for the sake of application compatibility is lost. Even traditional applications such as spreadsheets and word processors will require enough rethinking for the pen that the amount of rewriting will be substantial.

Apart from these reasons, using an existing operating system poses several other hurdles. Existing systems are designed for desktop use. They assume that the user is always plugged into a power source, and that any network connections are in place. Moreover, existing systems assume that unlimited disk space is available to support large applications. Mobile, pen-based computers, on the other hand, must be lightweight, compact, power-efficient and easily moved from one place to another.

For these reasons, GO saw no clear advantage in assembling an operating system from existing components.

It was clear that if GO built an operating system from the ground up, it could be expressly tailored to meet the needs of mobile, pen-based users. The system could be built around the unique benefits of the pen to achieve a new level of ease of use. In addition, the system could be optimized for mobile use and connectivity to a variety of other systems.

PenPoint: Designed for the Pen and Mobility

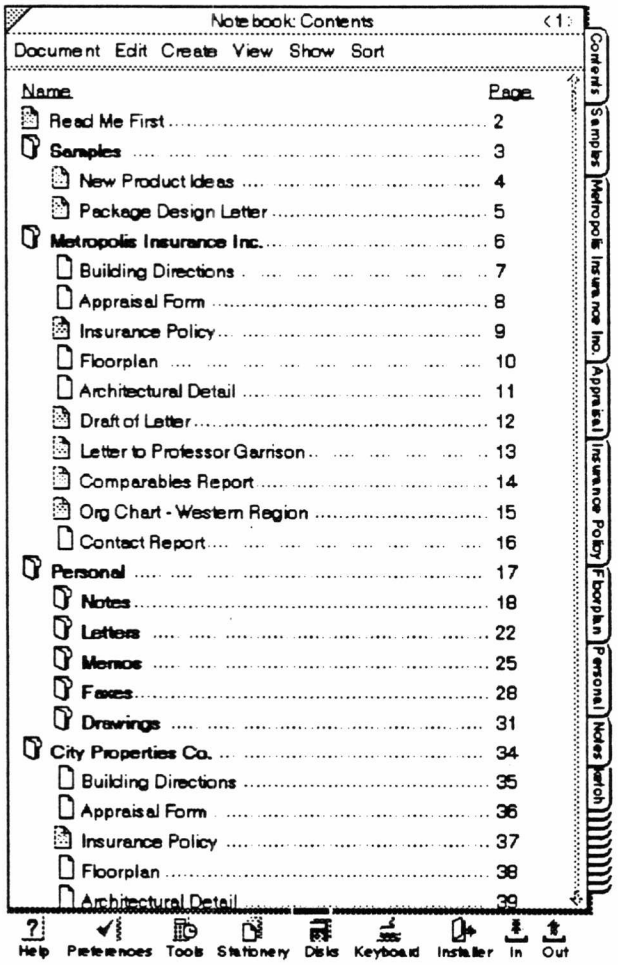
PenPoint is designed expressly to meet the needs of the mobile, pen-based computing market. It incorporates five key features that make possible a new class of computers that are easier to use than existing systems:

- **Notebook User Interface (NUI)** — A simple organizing metaphor — the Notebook — combined with a document model, commands (called "gestures") issued with a pen, and powerful handwriting translation.
- **Embedded Document Architecture (EDA™)** — PenPoint's EDA lets the user embed live, editable documents within one another and create hyperlink buttons for navigating through the Notebook.

- **Mobile Connectivity** — Instant-on, detachable networking and deferred I/O make possible truly portable computers for mobile users.
- **Compact and Scalable Implementation** — While expressly designed for small, lightweight, portable computers, PenPoint is highly hardware-independent and scales to a variety of formats, from pocket-size to wallboard-size computers.
- **Rich OS for the 1990s** — A true, 32-bit, multitasking, flat memory model architecture that supports the Intel 386 architecture and is readily portable to other processor families. PenPoint also provides a powerful and compact imaging model, ImagePoint™.

The NUI Delivers a New Level of Ease of Use

By combining intuitive organization and navigation with gestures and handwriting translation, PenPoint's Notebook User Interface, or NUI, delivers unparalleled ease of use.



Simple Navigation and Intuitive Organization

PenPoint uses a notebook metaphor to organize the user's work. Users are insulated from the complexities of applications and files, and interact instead with documents.

Like the notebook it imitates, PenPoint's NUI makes navigation simple and easy to learn. On the first page is a Table of Contents (TOC) that lists all of the documents in the Notebook. Associated with each document is a page number. Documents can be grouped into sections. Tabs can be attached to any document or section. To move from the TOC to a document, the user simply touches the appropriate page number in the TOC, the tab, or the document's icon.

The user can easily reorganize the contents of the TOC by dragging documents to different locations. The Notebook's pages are then automatically renumbered.

Below the TOC is the Bookshelf, where users can access often-used functions such as Help and Preferences.

A Table of Contents Instead of File Directories and Windows

Because the NUI and its TOC provide immediate access to every file in the computer, it is more appropriate for mobile, pen-based users than a desktop operating system's windows. With a traditional GUI, users must open every application and file they wish to use concurrently. The number of applications the system can accommodate is limited by the amount of memory available — usually only two or three.

But if the user wishes to open a new document, he or she must first decide if the appropriate application is running, and if not, select the application and load a file. Or, the user must go back to the desktop, find the correct file, attempt to load it, dismiss a different application if necessary to free up sufficient memory, and then retry to open the application. Once the user has the correct file open, it may need to be enlarged to full screen.

This process is acceptable in a desktop environment where the user's concentration is focused on the computer, but it does not meet the needs of mobile, pen-based users who require quick access to all of their data. PenPoint provides this access by allowing users to move easily from page to page.

To provide this capability, PenPoint implements the document model on top of a traditional file system. Switching from one page to another automatically starts the application on the new page and loads the appropriate data. Then, PenPoint dismisses the previous application after filing its data. All of this activity is hidden from the user.

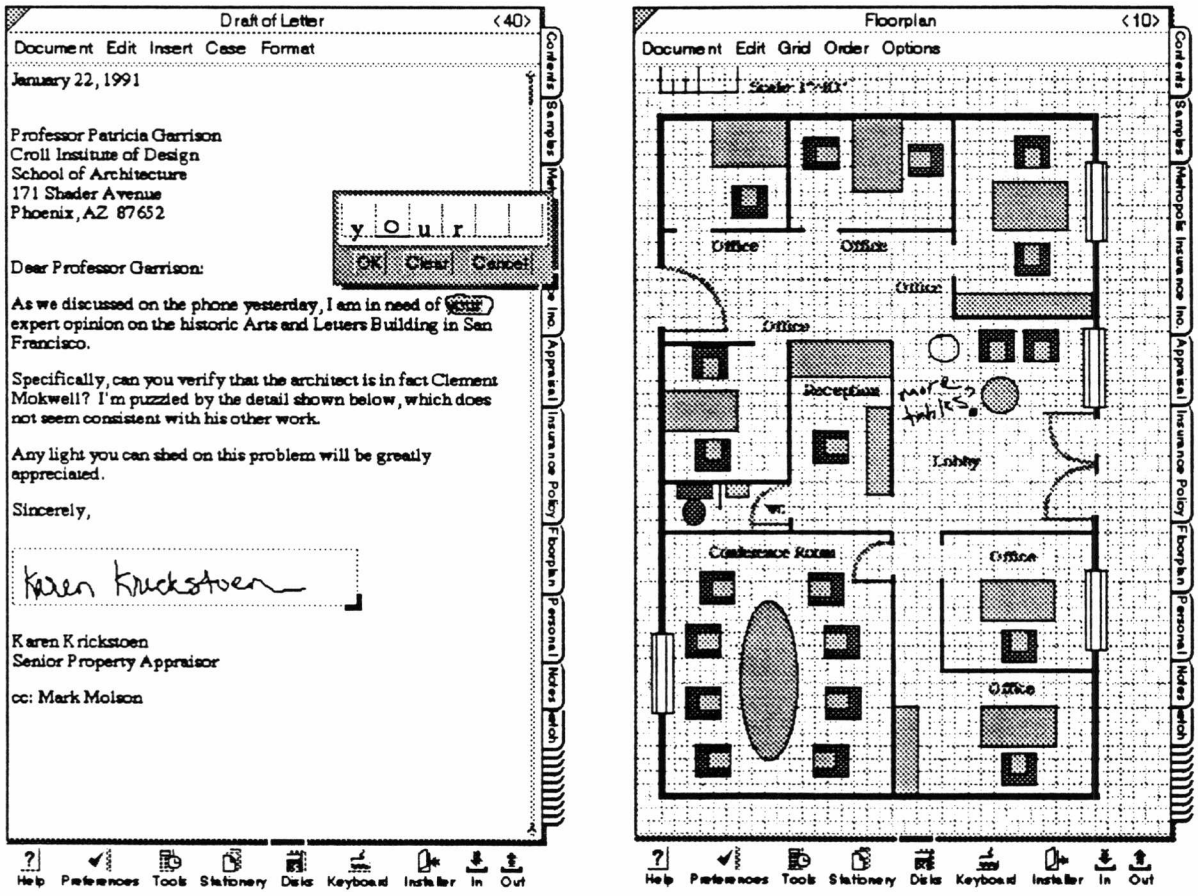
Gestures

Gestures are commands issued with a pen. They enable a pen to be significantly easier to use than a mouse and keyboard because they combine selection and action into a single, intuitive motion. Examples of gestures include crossing out words or graphics with an X, turning pages or scrolling with a flick, and moving text or graphics by dragging them with a pen. Users may also use the menu equivalents of gestures if they prefer.

Because PenPoint was designed for the pen from the ground up, its 11 basic gestures work consistently throughout the system and applications. The same gestures are used to edit text, control the Table of Contents or operate a drawing application. PenPoint users never fall back into a character-oriented world in which they must type onto a keyboard in order to modify a configuration or restart the environment, as they would in a desktop OS-based system.

In addition, PenPoint users don't have to press a button on the pen or the screen to differentiate between gestures and text. For example, a user can draw a circle on the screen in different applications, or even within the same application, and be assured of logically different results, depending on the context surrounding the circle.

In one case, the user might write a circle that will be translated as the letter "O," while in another instance, it will be interpreted as the gesture meaning "correct this word." In a third case, the user could draw a circle that PenPoint will interpret as a shape, while in a fourth instance, the drawn circle might be part of a signature.


Draft of Letter
Document Edit Insert Case Format
January 22, 1991
Professor Patricia Garrison
Croll Institute of Design
School of Architecture
171 Shader Avenue
Phoenix, AZ 87652
y o u r
OK Clear Cancel
Dear Professor Garrison:
As we discussed on the phone yesterday, I am in need of your expert opinion on the historic Arts and Letters Building in San Francisco.
Specifically, can you verify that the architect is in fact Clement Mokwell? I'm puzzled by the detail shown below, which does not seem consistent with his other work.
Any light you can shed on this problem will be greatly appreciated.
Sincerely,
Karen Krickstoen
Senior Property Appraisor
cc: Mark Molson
Floorplan
Document Edit Grid Order Options
Office
Reception
Lobby
Conference Room
Help Preferences Tools Stationery Disks Keyboard Installer In Out

PenPoint understands the difference between these four circles, allowing the freedom to "write" in a manner that comes naturally. This is possible because PenPoint applications supply contextual information that helps determine the meaning of a user's pen strokes, freeing the user from selecting a mode. This capability makes PenPoint easier to use than desktop-based systems and allows users to concentrate on their surroundings instead of their computers.

The Directness of the Pen

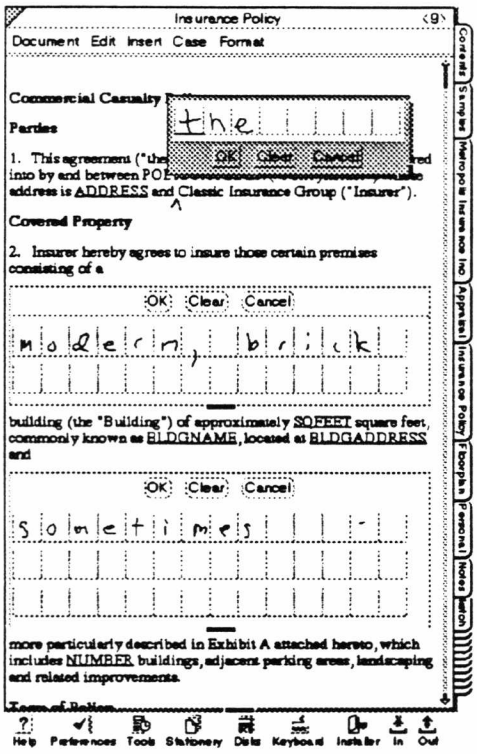
Under the PenPoint operating system, a user can touch the pen down wherever or whenever desired, and write onto as many areas of the screen as desired, just as one would with a pad of paper. In contrast, desktop operating systems assume that input comes from one place — the keyboard — and that input is directed to a single location chosen by the mouse or cursor keys.

PenPoint applications can also preserve a user's writing context when inserting text. For example, if a user adds a new sentence in the middle of a paragraph, the paragraph splits open so the user can see the preceding and subsequent sentences. Text is not obscured by input pads, nor must text always be entered into a reserved area on the screen.

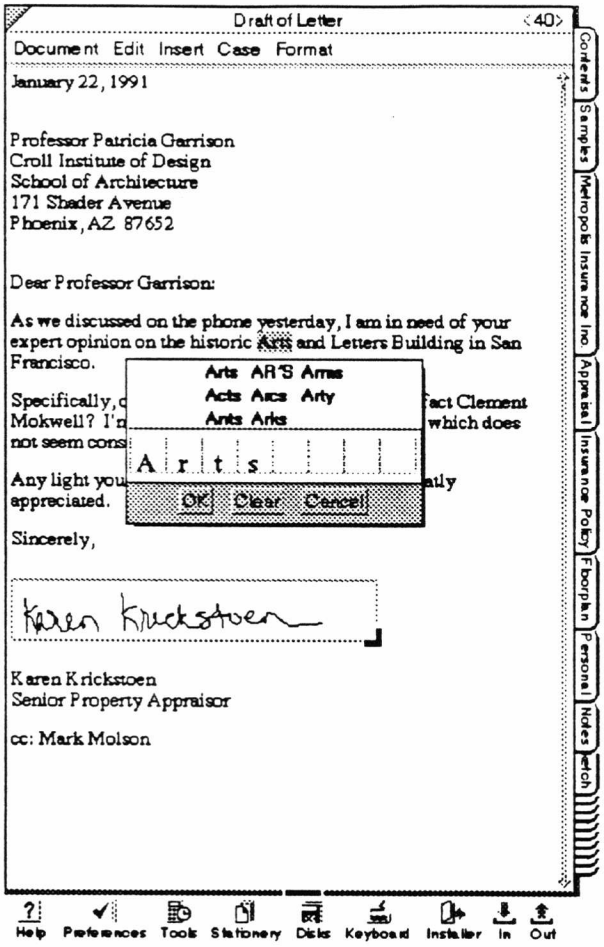
PenPoint users never need to drag a cursor around the screen, nor do they have to move a selection point or tap in a field before they can write. Cursors, insertion points and field selection are mouse and keyboard complexities that PenPoint users don't have to think about. Instead, they touch the pen to the screen and work.

Handwriting Translation

PenPoint's NUI provides a powerful handwriting translation system that accommodates a wide variety of handwriting styles. It is designed to reduce the amount of time required to input corrected text into a document. To achieve this goal, the NUI's handwriting translation system provides:


Insurance Policy
Document Edit Insert Case Format
Commercial Casualty
Parties
the
OK Clear Cancel
into by and between
address is ADDRESS and Classic Insurance Group ("Insurer").
Covered Property
2. Insurer hereby agrees to insure those certain premises consisting of a
OK Clear Cancel
molerm, brick
building (the "Building") of approximately SQFEET square feet, commonly known as BLDGNAME, located at BLDGADDRESS and
OK Clear Cancel
Sometimes
more particularly described in Exhibit A attached hereto, which includes NUMBER buildings, adjacent parking areas, landscaping and related improvements.
Help Preferences Tools Stationery Data Keyboard Installer In Out

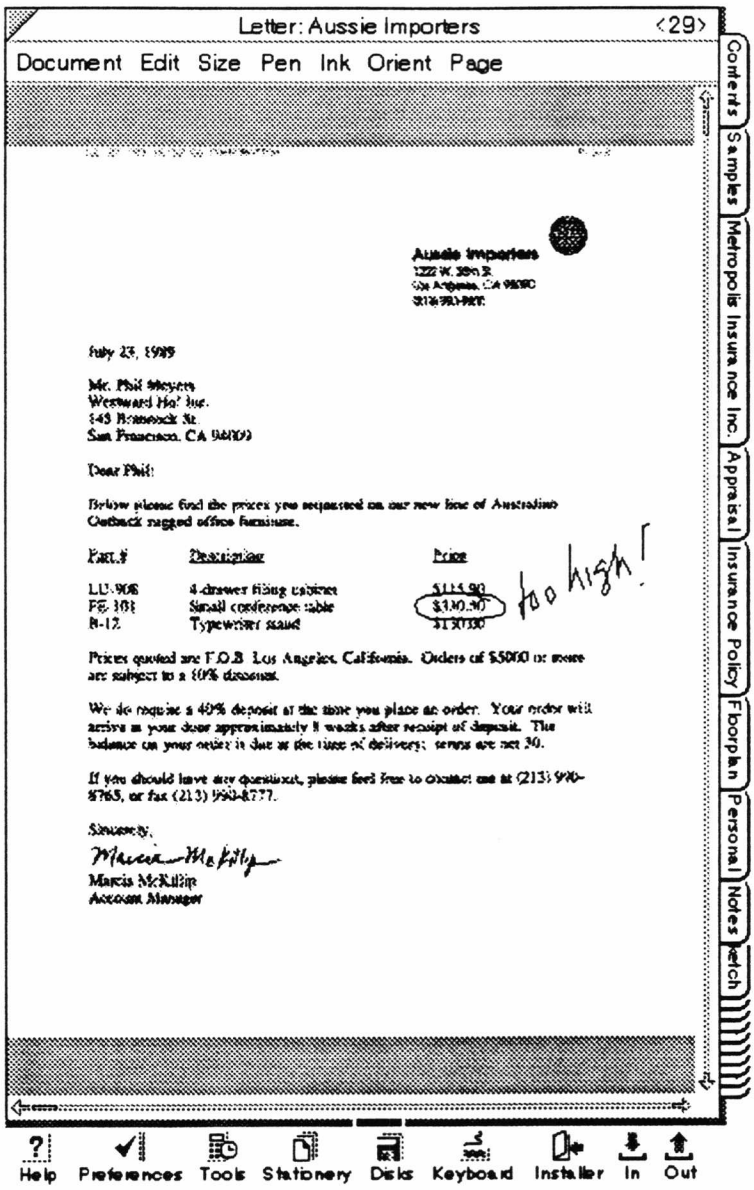
- Standard user interface components such as input pads that are tightly coupled with the translation system. After text is written into an input pad, it is translated and presented to the user in easy-to-correct editing pads. Then, when any necessary corrections are made, the user inserts the text into the document. This close coupling speeds the input of corrected text.
- The ability to accept upper and lower case characters, numerals, punctuation and some overlapping characters. The translation system is stroke order-independent, so users can dot an "i" or cross a "t" after they have finished writing a word.
- High levels of translation accuracy, measured not only by the percentage of characters translated correctly, but also by the percentage of words translated correctly. Word-level accuracy is important because it determines the number of corrections a user must make. High word-level accuracy translates into faster, more efficient text entry.
- A two-way link between applications and the translation system that allows applications to supply context to improve translation results. For example, an application can indicate whether it is expecting alpha or numeric input in a given field, or whether a circle will be interpreted as the letter "o" or an edit gesture.


Draft of Letter
Document Edit Insert Case Format
January 22, 1991
Professor Patricia Garrison
Croll Institute of Design
School of Architecture
171 Shader Avenue
Phoenix, AZ 87652
Dear Professor Garrison:
As we discussed on the phone yesterday, I am in need of your expert opinion on the historic Art and Letters Building in San Francisco.
Arts ARTS Arms
Acts Aics Arty
Ants Arks
A r t s
OK Clear Cancel
Sincerely,
Karen Krickstoen
Senior Property Appraisor
cc: Mark Molson
Help Preferences Tools Stationery Disks Keyboard Installer In Out

- The ability to present alternative translations so that users can select another if the first is incorrect.
- Case heuristics that capitalize words in a sentence appropriately. These heuristics also allow the 20 to 30 percent of users who prefer to write upper case to do so, while allowing their input to be translated correctly into the appropriate combination of upper and lower case characters.
- User-definable preferences that allow users to write in character boxes of various sizes or on ruled lines of various heights.
- The ability to train the handwriting system to recognize an individual's unique style of writing. Several users can share a single machine, easily switching between different handwriting profiles without re-booting or complex reconfiguration.

- Application Programming Interfaces (APIs) that allow other vendors to incorporate their own handwriting translation systems into PenPoint. This ensures that PenPoint users will always have access to the best available handwriting translation technology.

PenPoint's NUI also has the ability to preserve handwriting without translating it into text. This is particularly applicable where annotation or signature capture is required. For example, consider fax mark-up. A user might receive a fax on his PenPoint computer, make edits and comments in his own handwriting, and fax the marked-up document back to the sender.



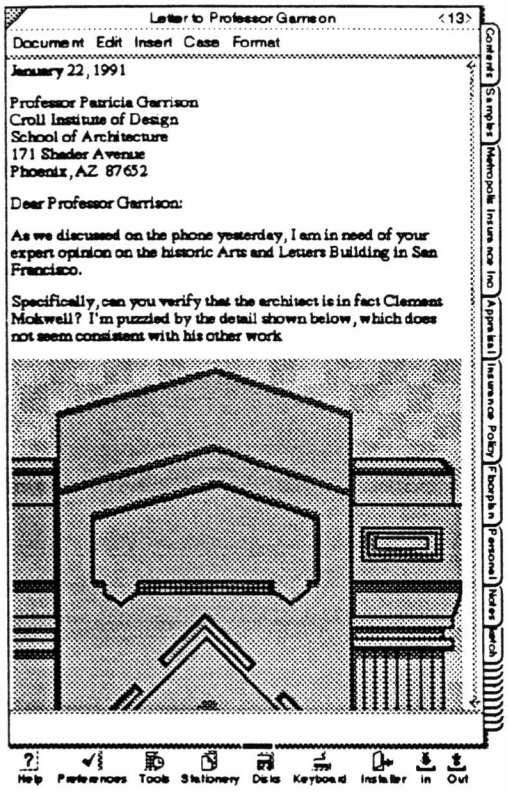
PenPoint also allows users to defer the translation of their handwriting into text, allowing them to preserve the visual context of their handwritten notes.

Embedded Document Architecture

PenPoint incorporates a design innovation called Embedded Document Architecture — EDA — that allows users to embed live documents inside of one another and to create hyperlinks between documents with a single gesture.

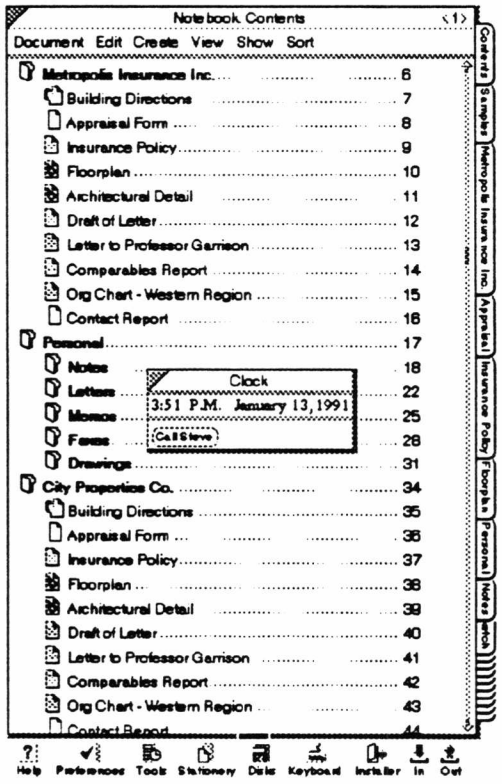
Live Document Embedding

EDA allows any PenPoint application to be embedded within any other to create compound documents. For example, the user can use a word processor to create a letter and embed a drawing into it. But, unlike traditional operating systems, both applications are "live" in the same document — the user can edit the letter or the embedded drawing at the same time. The user can even drag text from the letter into the drawing and vice versa.


Letter to Professor Garrison
Document Edit Insert Case Format
January 22, 1991
Professor Patricia Garrison
Croll Institute of Design
School of Architecture
171 Shader Avenue
Phoenix, AZ 87652
Dear Professor Garrison:
As we discussed on the phone yesterday, I am in need of your expert opinion on the historic Arts and Letters Building in San Francisco.
Specifically, can you verify that the architect is in fact Clement Mokwell? I'm puzzled by the detail shown below, which does not seem consistent with his other work.
Help
Preferences
Tools
Stationery
Disks
Keyboard
Installer
In
Out

Hyperlinks

EDA allows users to create hyperlink buttons that can be used to navigate between locations in the notebook. Users can even place hyperlink buttons in accessory applications such as clocks. Developers can use this capability to create fully configured Notebooks with predefined navigation paths. For example, a customer service Notebook might present its users with a series of buttons that lead to forms, customer service information — even a special help system. Yet EDA allows developers to provide this capability without writing any code. These hyperlink buttons work in embedded documents, too. In fact, any document — even a compound document — can be linked to any other.



PenPoint Delivers Mobility

PenPoint computers are truly mobile devices that can be used while walking, standing or kneeling. PenPoint is uniquely capable of meeting the needs of mobile users because it provides instant on, deferred input and output, detachable networking, and the ability to connect with a wide range of systems.

Instant On

Because a PenPoint computer can be consulted "on the fly," it can be used just like a physical notebook or organizer. Documents are available immediately because a PenPoint machine does not need to be re-booted when it is turned on. Booting is typically required only when a new version of the operating system is installed.

When PenPoint users turn their machine on, they are returned to the page that was being viewed when the computer was turned off. In contrast, desktop operating systems must be re-booted whenever the machine is turned on, and the applications and files must be re-loaded.

PenPoint's code is loaded into memory at boot time and stands ready to operate as soon as the processor awakens. The same is true of applications, which are loaded when they are installed. Starting and stopping PenPoint, then, is simple: The processor simply freezes, and all processes halt in place to begin again when the system "awakens."

If the PenPoint machine has backing store such as a hard disk, power-off causes the contents of memory to be copied to disk before the processor freezes. When power returns, the minimum required data is reloaded. All of this happens so quickly that the user experiences instant access.

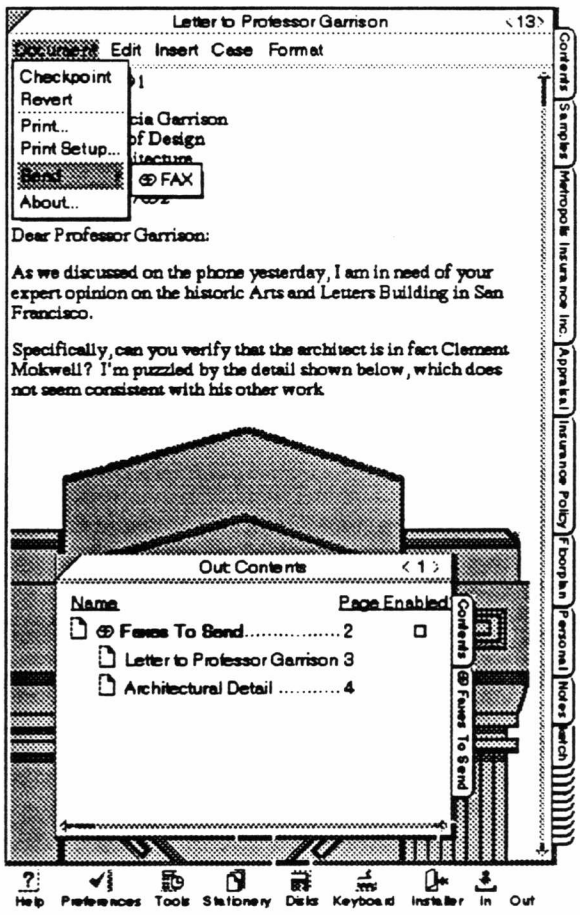
In contrast, desktop operating systems typically require code and data to be relocated into memory whenever they are powered on. The time required to load code and data prevents these machines from delivering instant-on capability.

PenPoint's instant-on capability is closely related to its power-saving technology. A PenPoint machine enjoys a long battery life because the processor and many of the hardware subsystems spend most of their time in a dormant state.

Deferred Input and Output

Even when they are away from any physical connection, PenPoint users can issue commands for printing, faxing, filing or any other form of data transfer. PenPoint simply moves a copy of the document to be transferred to the Out box, which is accessible from the Bookshelf. When the user makes the appropriate connection, the document is transferred out of the Out box automatically — without any assistance or intervention from the user.

Similarly, a PenPoint machine can awaken to request data from a host computer or receive a fax, even if it is left unattended. The incoming data is placed in the In box, also accessible from the Bookshelf. When the user returns to the machine, PenPoint informs the user that documents have arrived in the In box, which the user can subsequently check.



This capability, called Deferred I/O, is a critical feature for mobile users because it frees them from having to remain connected to outside services. It is accomplished through a set of connectivity Application Programming Interfaces (APIs). These APIs provide communications protocols, automatic detection of connections (to networks, for example), support for direct access to files formatted for a variety of operating systems, translation of files to and from PenPoint internal formats, deferred I/O and easy access to, and management of, connected devices.

Detachable Connectivity

PenPoint users can connect and disconnect from networks without re-booting or performing special manipulations. The user simply unplugs the network cable, and network operations are suspended. Later, when the cable is reconnected, network operations are resumed and any documents waiting in the Out box can be transferred.

In addition, PenPoint users can switch between different networks by simply plugging in the relevant cables. This is possible because several protocols can be installed in a PenPoint machine simultaneously. When PenPoint senses that a network connection has been attempted, it awakens the appropriate protocol and resumes the connection.

Finally, PenPoint is ideal for use with wireless network connections because it will not lose data when sessions are interrupted by network interference. PenPoint can simply suspend and resume the connection as a user's PenPoint machine moves in and out of the network range.

Connecting PenPoint to Other Systems

Because mobile, pen-based computers must access data stored in many places, PenPoint is designed to connect with a wide variety of systems.

- PenPoint can use a special version of TOPS networking software to move, copy, and manipulate files on PCs and Macintoshes, and any network volumes available to them.
- PenPoint allows other network protocols to be installed and configured on the fly. GO is currently working with Novell to provide a Netware client for PenPoint.
- PenPoint provides an extensible file system that can be adapted to support any operating system's file format. In its first implementation, PenPoint reads and writes MS-DOS formatted disks.
- PenPoint provides a consistent, easy-to-use import and export system. When a user attempts to copy a foreign data file into PenPoint, a dialog appears which lists the document types (applications) that will accept the file.

  For example, if a user copied a Lotus 1-2-3 file called BUDGET.WK1 into PenPoint, the dialog would list the installed applications that accept 1-2-3 files. The user might select a spreadsheet, in which case a new document would appear in the table of contents called "Budget." When the user selected that page, the Budget spreadsheet would appear.

  Export is equally simple. The user selects the desired file format, and the application performs the translation. The file is placed on the user's floppy disk or in a directory on an attached PC.

PenPoint is Compact and Scalable

The mobile, pen-based market demands different sizes of computers that can run the same applications. PenPoint's compact and scalable design supports these diverse configurations, yet its hardware-independent APIs allow the same applications to run unmodified on any PenPoint machine.

PenPoint machines can:

- Vary greatly in format, from shirt-pocket to wallboard-size. In fact, a PenPoint machine could be designed to sit directly on any overhead projector, allowing it to serve as an electronic acetate.
- Use many types of pens, tethered or untethered; proximity sensing or not.
- Provide single- or dual-tier memory architectures.
- Include or not include a floppy disk drive.
- Include cellular modems, fax modems, conventional modems, etc.

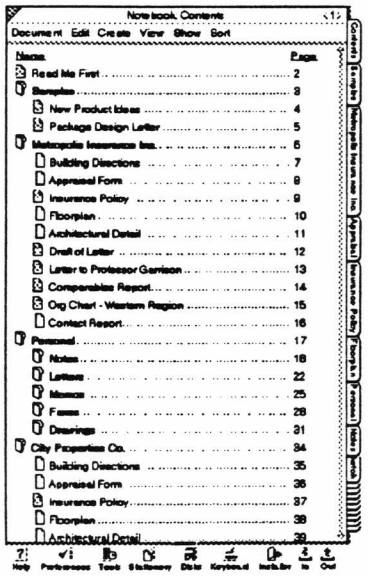
Device Independence

PenPoint provides this unique opportunity for hardware differentiation because its hardware dependencies are isolated in a small portion of PenPoint called the Machine Interface Layer, or MIL. The MIL connects PenPoint to hardware. When PenPoint is ported to new hardware, only the MIL must be rewritten.

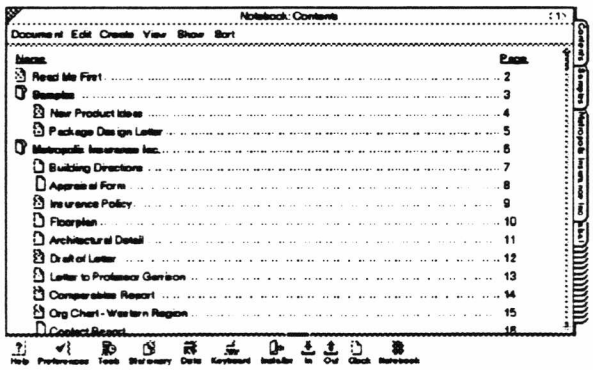
Since PenPoint applications don't communicate directly with hardware, the same applications run unmodified on any PenPoint machine.

Scalable User Interface

PenPoint's user interface components provide a device-independent coordinate system and auto-layout capability that allow applications to present themselves attractively regardless of screen size, resolution or aspect ratio.



PenPoint's auto-layout capability allows developers to set parameters for user interface components in relative terms. For example, in the table of contents, the page numbers and tabs are



positioned relative to the right side of the screen, while the document titles are positioned with respect to the left. When the screen orientation is switched from landscape to portrait, the table of contents still looks logical. This same capability allows PenPoint applications to "look right" on any size screen — from shirt-pocket to wallboard-size.

No Keyboard Required

Since PenPoint machines do not require a keyboard, they can be significantly smaller than keyboard-based portables. Of course, PenPoint accepts keyboard input and even provides an on-screen, "virtual" keyboard that allows users to tap out characters if they wish.

Memory Conservation

PenPoint requires less memory than a desktop operating system with a pen because PenPoint was designed with low memory consumption in mind. In contrast, reduced memory consumption was not a key design criteria for desktop operating systems.

Reduced memory requirements mean fewer memory chips and smaller computers. In contrast to desktop GUI applications, which typically require multiple megabytes of memory, PenPoint applications typically require only 100 to 200 kilobytes.

PenPoint conserves memory by using a single copy of code instead of requiring one copy in memory and another on disk, as is required by desktop operating systems. PenPoint also allows code to be shared. Its object-oriented design allows developers to inherit most of their applications' behavior from components included in PenPoint. Because these objects are re-entrant and re-usable, multiple PenPoint applications do not require multiple copies of code.

Different Memory Configurations

PenPoint supports different physical memory architectures, from all-DRAM to a combination of dynamic random access memory (DRAM), static random access memory (SRAM), and rotating or silicon hard disks. Each of these combinations meets a particular market need.

- All DRAM. Suitable for small, inexpensive PenPoint systems where small form factor and/or low hardware costs are important.
- SRAM and FLASH. Suitable for small PenPoint systems in which protection of data against power loss for sustained periods of time is important. FLASH memory does not lose data when the system loses power.
- DRAM and backing store. Suitable for PenPoint users who require rapid access to large amounts of data.

An Operating System for the 1990s

Because PenPoint is built on a strong foundation of advanced technologies, it will serve the mobile, pen-based market through the 1990s and beyond. These technologies include:

- Object-oriented interfaces down to the kernel. Developers can make use of object-oriented interfaces for every function that PenPoint provides.
- True 32-bit architecture. PenPoint's 32-bit addresses and flat memory model take full advantage of the 386 processor and provide PenPoint with the opportunity to be ported to other processor families.
- Pre-emptive multitasking. PenPoint's multitasking capabilities are similar to OS/2's, enabling smooth user interface interactions, background communications and smooth translation of handwriting while the user is writing.
- A compact and powerful imaging model. PenPoint's ImagePoint imaging model unifies screen display and printing. All text operations are fully integrated with graphics operations. Outline fonts are used to render text on demand at any size. In addition, ImagePoint provides scaling, translation, rotation and sampled image rendering.
- Extensible and replaceable modules, including the imaging model, the handwriting translation system and all of the services.
- Automatic installation of applications. Users don't have to deal with complicated installation scripts. Instead, they just insert a disk into a drive, and the PenPoint Installer handles the rest.
- Dynamic binding. Applications can leverage software components provided with other applications by binding with them at installation time.

Summary

The potential of pen-based computing is best realized with an operating system designed from the ground up to meet the special requirements of mobility and the pen. In this way, applications can fully exploit the capabilities of the pen while simplifying what the user needs to know to operate a mobile, pen-based computer.

PenPoint meets these requirements with the five innovations described in this paper:

- The Notebook User Interface, or NUI, that combines a notebook metaphor with gestural control and handwriting translation;
- The Embedded Document Architecture, or EDA, that allows users to create linked, compound documents;
- A compact and scalable design that allows PenPoint machines to take on a wide variety of physical characteristics;

- Features for mobile connectivity that include detachable networking, simultaneous installation and residence of multiple protocols, deferred I/O, and an easy-to-use import and export facility; and,
- A combination of advancements including object-oriented interfaces and a highly portable, 32-bit, flat memory model design that makes PenPoint an operating system for the 1990s and beyond.

With PenPoint, users have a natural way to apply the power of computing in new situations, developers have a modern operating system that they can use with existing tools and skills, and hardware companies have the opportunity to build exciting, differentiable products.

# # #

The GO and PenPoint logos and names are trademarks of GO Corp., EDA and ImagePoint are trademarks of GO Corp. All other marks are trademarks or registered trademarks of the manufacturers with which the marks are associated.

Glossary

Automatic constraint-based layout:	A mechanism for describing the design of graphical user interface elements in relative rather than absolute terms. This allows the user interface to adapt automatically to display characteristics such as resolution, size and aspect ratio.
Bookshelf:	Found at the bottom of the PenPoint screen, the bookshelf contains icons that represent PenPoint accessories.
Deferred I/O:	Documents and files to be transmitted via E-mail or fax or to be printed are queued until the appropriate connections are made.
Detachable networking:	The ability to disconnect and reconnect to networks without the need for special commands or re-booting.
Document model:	A feature of NUI that overlays a document model on top of a traditional file system, allowing users to deal with documents rather than with applications and files. All documents are immediately available, and the user can easily switch between them by simply turning pages in the Notebook.
Editing Pad:	A pad containing text the user wants to change.
Embedded Document Architecture (EDA):	A design innovation that allows users to embed live documents inside each other and to create hyperlinks between and within documents.
Flick:	A gesture that is used to scroll documents and move between pages in the Notebook.
Handwriting profile:	A customized handwriting prototype set created and saved when a user runs the Handwriting Customization accessory.
Hyperlink:	A button that is created by the user or programmer to allow automatic navigation to another location in the Notebook or within the same document.
ImagePoint:	A compact and powerful imaging model that provides scaling, translation, rotation, bezier curves, sampled image rendering and outline fonts.
In box:	A system service that stores incoming network transmissions such as E-mail, faxes and file transfers.

Input pad:	An area on the screen into which the user enters handwriting. Input pads come in various sizes and types, such as boxed or ruled.
MIL:	The Machine Interface Layer. A portion of PenPoint that isolates hardware-specific code.
Out box:	A system service that stores outgoing transmissions such as E-mail, faxes and printing tasks until the appropriate connection is established.
Notebook:	The structure containing documents, applications and information in PenPoint.
Notebook Table of Contents:	Lists all of the sections and documents in the Notebook.
NUI:	The Notebook User Interface provides a simple organizing concept (the Notebook metaphor), a document model, gestural commands and a powerful handwriting translation system.
Pre-emptive multitasking:	An approach to multitasking that allows the operating system to determine when and to whom system cycles are allocated. This is in contrast to "cooperative multitasking," in which the applications themselves determine when they will release the resources they are using.
Tab:	A marker for a document or section in the Notebook. The user taps a tab to go to that page.
Tap:	The user puts the pen down on the screen and lifts it away. The tap is used to open menus, choose commands, turn to pages and select some items in the Notebook.
TOPS:	A networking protocol from SITKA Corp. which, when installed on a PenPoint system, allows a user to move, copy and manipulate files on PC and Macintosh computers.